



Aula 10 - ESP32 - Configuração do Acesso Bluetooth



Módulo de Internet das Coisas

Prof.



UNIVERSIDADE
ESTADUAL DO CEARÁ



MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO





IA

Objetivos da Aula

- Configurar um servidor BLE no ESP32
- Configurar um scanner BLE no ESP32
- Testar comunicação BLE entre ESP32 e smartphone
- Modificar o exemplo Notify com leituras do PIR Sensor



IA

Demo-Lab 1 - Configurar um servidor BLE no ESP32 [1]

Na Arduino IDE, **Files>Examples>ESP32 BLE Arduino**, selecionar o exemplo BLE Server. Então, seguir os seguintes passos para criar o BLE server no ESP32:

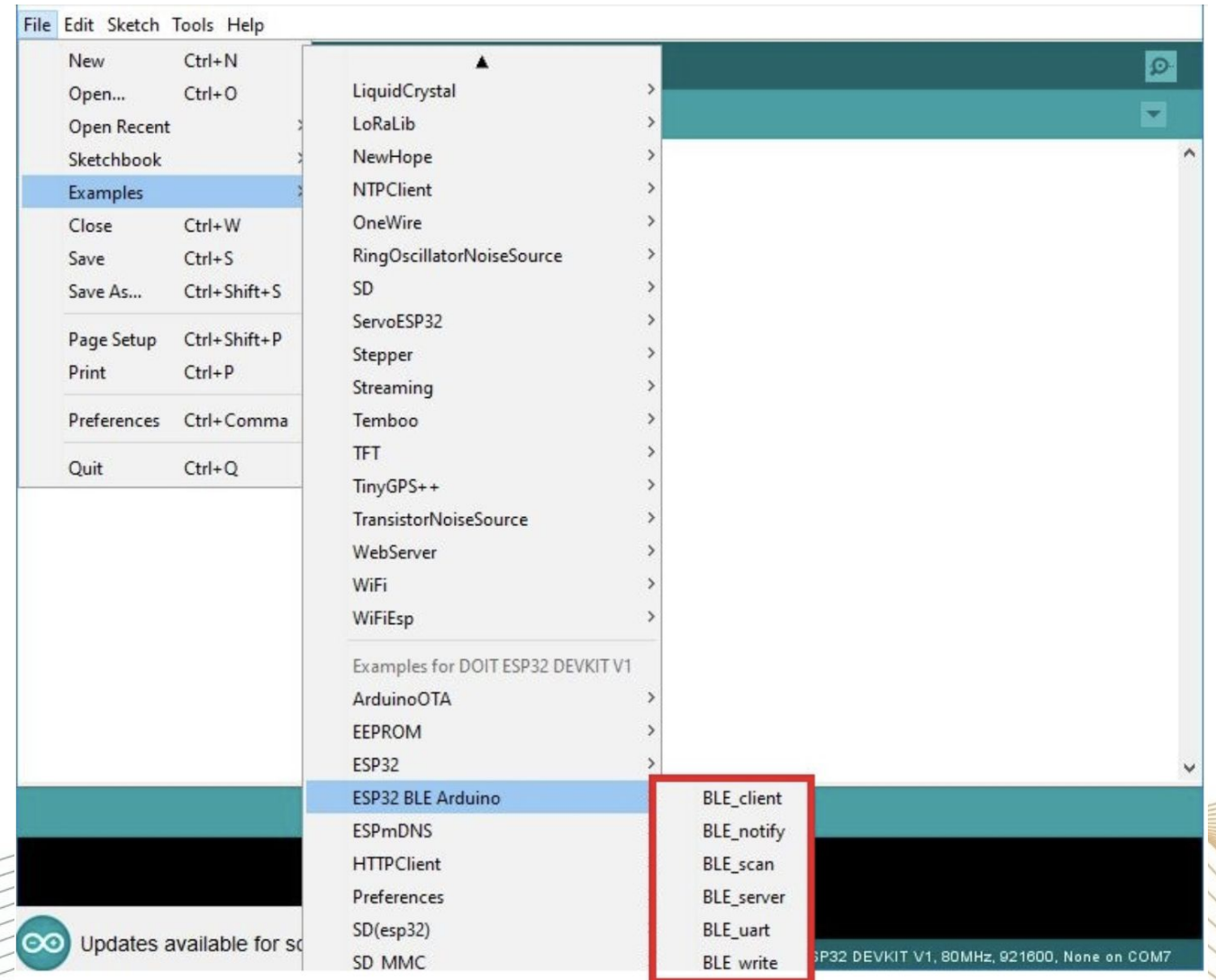
1. Criar um BLE service
2. Criar uma configuração/característica para o serviço BLE
3. Criar um BLE Descriptor de uma configuração
4. Iniciar o Serviço
5. Iniciar o anúncio, para que seu ESP32 seja encontrado por outros dispositivos



IA

Demo-Lab 1 - Configurar um servidor BLE no ESP32 [1]

O Arduino IDE traz uma série de códigos que testam o BLE do ESP32:
File> Examples> ESP32 BLE Arduino.





1 - Criar um servidor BLE

- Abra o Arduino IDE, File> Example> ESP32 BLE Arduino> BLE Server
- O código segue a seguinte sequência de criação do servidor:
 - Cria o BLE Server
 - Cria um BLE Service
 - Cria uma característica no BLE Service
 - Cria um BLE Descriptor na Característica
 - Inicia o Serviço
 - Inicia o anúncio para que outros dispositivos o encontrem.



- o Código inicia com a importação de bibliotecas do BLE

```
#include <BLEDevice.h>
#include <BLEUtils.h>
#include <BLEServer.h>
```

- Em seguida, define-se os UUIDs para característica e serviço

```
#define SERVICE_UUID "4fafc201-1fb5-459e-8fcc-c5c9c331914b"
#define CHARACTERISTIC_UUID "beb5483e-36e1-4688-b7f5-ea07361b26a8"
```



IA

Demo-Lab 1 - Configurar um servidor BLE no ESP32 [1]

No setup():

- o Serial Monitor é inicializado

```
Serial.begin(115200);
```

- O dispositivo "MyESP32" é criado. Modifique esse valor com seu nome.

```
// Create the BLE Device  
BLEDevice::init("MyESP32");
```

- o dispositivo é configurado como servidor

```
BLEServer *pServer = BLEDevice::createServer();
```

- e o UUID do servidor é configurado

```
BLEService *pService = pServer->createService(SERVICE_UUID);
```



No setup():

- A característica do serviço é configurada com propriedades de leitura e escrita

```
BLECharacteristic *pCharacteristic = pService->createCharacteristic(  
    CHARACTERISTIC_UUID,  
    BLECharacteristic::PROPERTY_READ |  
    BLECharacteristic::PROPERTY_WRITE  
);
```

- O valor da característica é configurado. Geralmente, o valor é uma leitura de um sensor ou o estado de um componente do dispositivo inteligente

```
pCharacteristic->setValue("Hello World says Neil");
```




No setup ():

- No final, o serviço é inicializado com a função de anúncio para que outros dispositivos BLE o encontrem

```
BLEAdvertising *pAdvertising = pServer->getAdvertising();  
pAdvertising->start();
```



IA

Demo-Lab 2 - Configurar um scanner no ESP32 [1]

O professor deve executar isso aqui primeiro.

1. Na Arduino IDE, **Files>Examples>ESP32 BLE Arduino**, selecionar o exemplo **BLE_scan**.
2. Fazer o upload do código BLE_scan para o ESP32
3. Em outro ESP32, fazer o upload do BLE Server
4. No ESP scanner, aperte o botão ENABLE para reiniciar e espere alguns segundos enquanto ele realiza o scan
5. Verifique se ele encontra o MyESP32 dos alunos

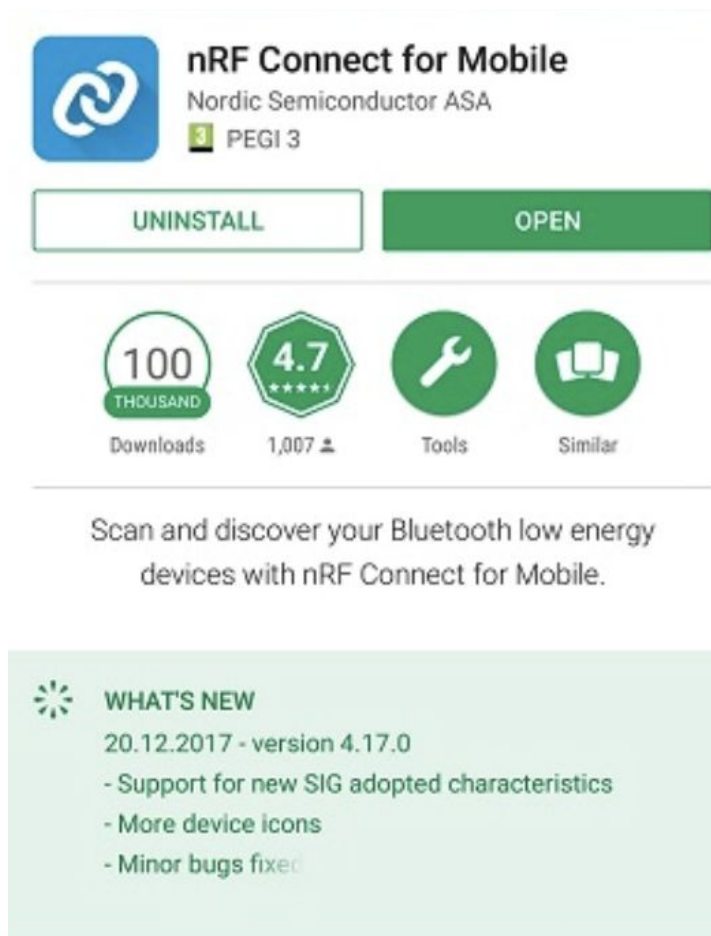
Agora, os alunos executam o scanner e o professor o servidor.



IA

Demo-Lab 3 - Comunicação BLE entre ESP32 e smartphone [1]

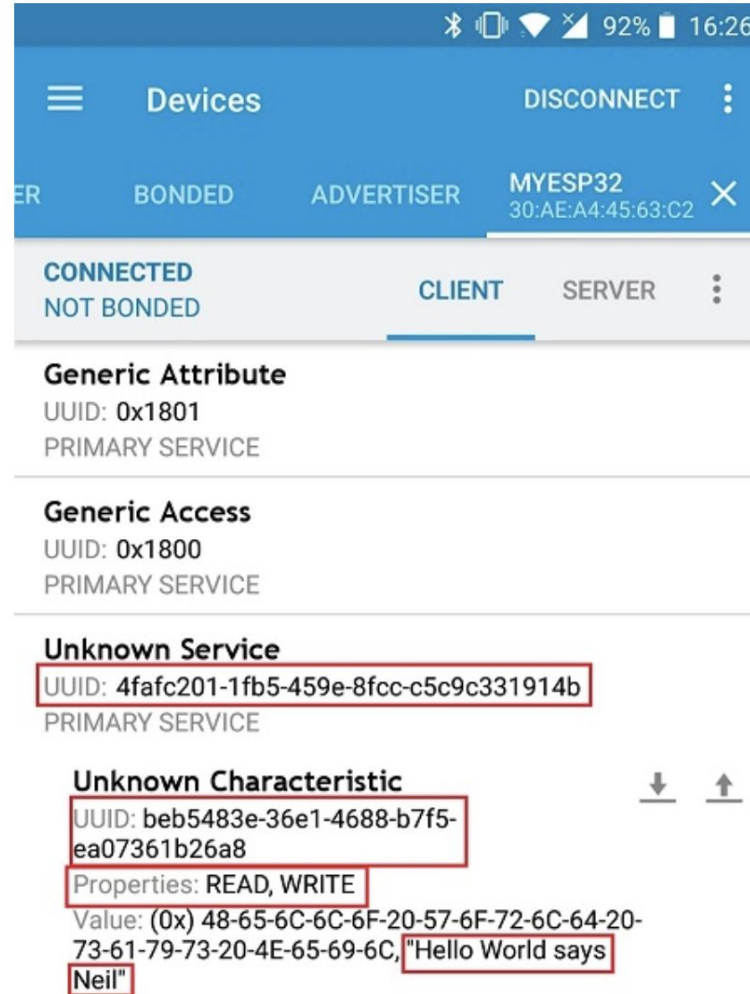
1. No seu smartphone, instalar a app nRF Connect for Mobile da Nordic: Android (Google Play Store) e iOS (App Store)
2. No seu Smartphone, habilitar o Bluetooth e visibilidade/localização.
3. Faça o upload do código BLE server no ESP32
4. Conecte seu smartphone com o ESP32 encontrado "ESP32-Seunome"





IA

Demo-Lab 3 - Comunicação BLE entre ESP32 e smartphone [1]





IA

Demo-Lab 4 - Modificar exemplo notify com o PIR sensor

1. Na Arduino IDE, **Files>Examples>ESP32 BLE Arduino**, selecionar o exemplo **BLE_notify**.
2. Modificar o exemplo para o ESP32 adicionando código de leitura do PIR sensor para o ESP32 notificar presença ao smartphone



IA

Referências Bibliográficas

[1] Rui Santos. Getting started with ESP32 Bluetooth Low Energy (BLE) on Arduino IDE. 2019. Disponível em:
<<https://randomnerdtutorials.com/esp32-bluetooth-low-energy-ble-arduino-ide/>>. Acesso em 06/04/2023.

Dúvidas?

Módulo de Internet das Coisas



UNIVERSIDADE
ESTADUAL DO CEARÁ



Instituto Iracema
PESQUISA E INOVAÇÃO



MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO

